**Міністерство освіти і науки України**

**Тернопільський національний технічний університет ім. Івана Пулюя**

Факультет інформаційних систем та програмної інжинерії

Кафедра кібербезпеки

**ЗВІТ**

до лабораторної роботи № 6

з дисципліни «Програмування мовою JAVA»

на тему:

**Застосування класів-колекцій і класів утиліт.**

Виконав студент групи СБ-21

Антонов Роман Олександрович

Перевірив:

Луцків Андрій Мирославович

Тернопіль, 2024

Посилання на репозиторій:

<https://github.com/rom341/JavaLabsTNTU>

Результат виконання:

===============================================

HashSet Demonstration

Proving, that HashCode generates same value for same objects:

Entity: ExampleEntity{Name='Eva', Age=94}

First gen: 2173039

Second gen: 2173039

Entity created: ExampleEntity{Name='Michael', Age=5} With HashCode: -1610628449

Entity created: ExampleEntity{Name='Tom', Age=33} With HashCode: 2613488

Entity created: ExampleEntity{Name='Diana', Age=14} With HashCode: 2046751392

Entity created: ExampleEntity{Name='Alina', Age=36} With HashCode: 1963872852

Entity created: ExampleEntity{Name='Tom', Age=80} With HashCode: 2613535

Entity created: ExampleEntity{Name='Tom', Age=92} With HashCode: 2613547

Entity created: ExampleEntity{Name='Tom', Age=76} With HashCode: 2613531

Entity created: ExampleEntity{Name='Oleg', Age=77} With HashCode: 76276879

Entity created: ExampleEntity{Name='Oleg', Age=4} With HashCode: 76276806

Entity created: ExampleEntity{Name='Michael', Age=82} With HashCode: -1610628372

HashSet values:

[ExampleEntity{Name='Michael', Age=5}, ExampleEntity{Name='Michael', Age=82}, ExampleEntity{Name='Oleg', Age=77}, ExampleEntity{Name='Tom', Age=33}, ExampleEntity{Name='Tom', Age=80}, ExampleEntity{Name='Alina', Age=36}, ExampleEntity{Name='Tom', Age=92}, ExampleEntity{Name='Tom', Age=76}, ExampleEntity{Name='Oleg', Age=4}, ExampleEntity{Name='Diana', Age=14}]

===============================================

===============================================

TreeSet Demonstration

Entity created: ExampleEntity{Name='Alina', Age=79}

Entity created: ExampleEntity{Name='Tom', Age=10}

Entity created: ExampleEntity{Name='Eva', Age=8}

Entity created: ExampleEntity{Name='Michael', Age=34}

Entity created: ExampleEntity{Name='Oleg', Age=96}

Entity created: ExampleEntity{Name='Diana', Age=63}

Entity created: ExampleEntity{Name='Denis', Age=28}

Entity created: ExampleEntity{Name='Viktoria', Age=34}

Entity created: ExampleEntity{Name='Eva', Age=56}

Entity created: ExampleEntity{Name='Michael', Age=28}

Generated TreeSet:

[ExampleEntity{Name='Oleg', Age=96}, ExampleEntity{Name='Alina', Age=79}, ExampleEntity{Name='Diana', Age=63}, ExampleEntity{Name='Eva', Age=56}, ExampleEntity{Name='Michael', Age=34}, ExampleEntity{Name='Denis', Age=28}, ExampleEntity{Name='Tom', Age=10}, ExampleEntity{Name='Eva', Age=8}]

compareTo demonstration:

(comparing by age)

ExampleEntity{Name='Oleg', Age=96} compareTo ExampleEntity{Name='Alina', Age=79}

Result: -1

===============================================

===============================================

TreeMap Demonstration

Entity created: Key: 0 Value: ExampleEntity{Name='Violeta', Age=37}

Entity created: Key: 1 Value: ExampleEntity{Name='Denis', Age=20}

Entity created: Key: 2 Value: ExampleEntity{Name='Viktoria', Age=32}

Entity created: Key: 3 Value: ExampleEntity{Name='Jackson', Age=0}

Entity created: Key: 4 Value: ExampleEntity{Name='Jackson', Age=68}

Entity created: Key: 5 Value: ExampleEntity{Name='Jackson', Age=46}

Entity created: Key: 6 Value: ExampleEntity{Name='Jackson', Age=65}

Entity created: Key: 7 Value: ExampleEntity{Name='Tom', Age=5}

Entity created: Key: 8 Value: ExampleEntity{Name='Violeta', Age=23}

Entity created: Key: 9 Value: ExampleEntity{Name='Oleg', Age=45}

Generated TreeMap:

{0=ExampleEntity{Name='Violeta', Age=37}, 1=ExampleEntity{Name='Denis', Age=20}, 2=ExampleEntity{Name='Viktoria', Age=32}, 3=ExampleEntity{Name='Jackson', Age=0}, 4=ExampleEntity{Name='Jackson', Age=68}, 5=ExampleEntity{Name='Jackson', Age=46}, 6=ExampleEntity{Name='Jackson', Age=65}, 7=ExampleEntity{Name='Tom', Age=5}, 8=ExampleEntity{Name='Violeta', Age=23}, 9=ExampleEntity{Name='Oleg', Age=45}}

Value with key='3': ExampleEntity{Name='Jackson', Age=0}

===============================================

===============================================

LinkedList Demonstration

Entity created: ExampleEntity{Name='Diana', Age=95}

Entity created: ExampleEntity{Name='Jackson', Age=8}

Entity created: ExampleEntity{Name='Michael', Age=95}

Entity created: ExampleEntity{Name='Tom', Age=86}

Entity created: ExampleEntity{Name='Alina', Age=22}

Entity created: ExampleEntity{Name='Alina', Age=51}

Entity created: ExampleEntity{Name='Oleksandra', Age=60}

Entity created: ExampleEntity{Name='Jackson', Age=0}

Entity created: ExampleEntity{Name='Violeta', Age=22}

Entity created: ExampleEntity{Name='Alina', Age=89}

Generated LinkedList:

[ExampleEntity{Name='Diana', Age=95}, ExampleEntity{Name='Jackson', Age=8}, ExampleEntity{Name='Michael', Age=95}, ExampleEntity{Name='Tom', Age=86}, ExampleEntity{Name='Alina', Age=22}, ExampleEntity{Name='Alina', Age=51}, ExampleEntity{Name='Oleksandra', Age=60}, ExampleEntity{Name='Jackson', Age=0}, ExampleEntity{Name='Violeta', Age=22}, ExampleEntity{Name='Alina', Age=89}]

===============================================

===============================================

ArrayList Demonstration

Entity created: ExampleEntity{Name='Roman', Age=44}

Entity created: ExampleEntity{Name='Roman', Age=40}

Entity created: ExampleEntity{Name='Tom', Age=85}

Entity created: ExampleEntity{Name='Alina', Age=23}

Entity created: ExampleEntity{Name='Violeta', Age=93}

Entity created: ExampleEntity{Name='Eva', Age=49}

Entity created: ExampleEntity{Name='Tom', Age=82}

Entity created: ExampleEntity{Name='Michael', Age=67}

Entity created: ExampleEntity{Name='Diana', Age=0}

Entity created: ExampleEntity{Name='Oleksandra', Age=15}

Generated ArrayList:

[ExampleEntity{Name='Roman', Age=44}, ExampleEntity{Name='Roman', Age=40}, ExampleEntity{Name='Tom', Age=85}, ExampleEntity{Name='Alina', Age=23}, ExampleEntity{Name='Violeta', Age=93}, ExampleEntity{Name='Eva', Age=49}, ExampleEntity{Name='Tom', Age=82}, ExampleEntity{Name='Michael', Age=67}, ExampleEntity{Name='Diana', Age=0}, ExampleEntity{Name='Oleksandra', Age=15}]

===============================================

===============================================

Queue Demonstration

Entity created: ExampleEntity{Name='Viktoria', Age=12}

Entity created: ExampleEntity{Name='Oleg', Age=82}

Entity created: ExampleEntity{Name='Tom', Age=11}

Entity created: ExampleEntity{Name='Denis', Age=26}

Entity created: ExampleEntity{Name='Roman', Age=46}

Entity created: ExampleEntity{Name='Diana', Age=9}

Entity created: ExampleEntity{Name='Michael', Age=82}

Entity created: ExampleEntity{Name='Eva', Age=15}

Entity created: ExampleEntity{Name='Viktoria', Age=74}

Entity created: ExampleEntity{Name='Viktoria', Age=71}

Entity dequeued: ExampleEntity{Name='Viktoria', Age=12}

Entity dequeued: ExampleEntity{Name='Oleg', Age=82}

Entity dequeued: ExampleEntity{Name='Tom', Age=11}

Entity dequeued: ExampleEntity{Name='Denis', Age=26}

Entity dequeued: ExampleEntity{Name='Roman', Age=46}

Entity dequeued: ExampleEntity{Name='Diana', Age=9}

Entity dequeued: ExampleEntity{Name='Michael', Age=82}

Entity dequeued: ExampleEntity{Name='Eva', Age=15}

Entity dequeued: ExampleEntity{Name='Viktoria', Age=74}

Entity dequeued: ExampleEntity{Name='Viktoria', Age=71}

===============================================

===============================================

Queue Demonstration

Entity created: ExampleEntity{Name='Viktoria', Age=21}

Entity created: ExampleEntity{Name='Denis', Age=70}

Entity created: ExampleEntity{Name='Tom', Age=84}

Entity created: ExampleEntity{Name='Eva', Age=35}

Entity created: ExampleEntity{Name='Violeta', Age=57}

Entity created: ExampleEntity{Name='Tom', Age=86}

Entity created: ExampleEntity{Name='Violeta', Age=87}

Entity created: ExampleEntity{Name='Diana', Age=21}

Entity created: ExampleEntity{Name='Jackson', Age=76}

Entity created: ExampleEntity{Name='Oleksandra', Age=85}

Entity dequeued: ExampleEntity{Name='Violeta', Age=87}

Entity dequeued: ExampleEntity{Name='Tom', Age=86}

Entity dequeued: ExampleEntity{Name='Oleksandra', Age=85}

Entity dequeued: ExampleEntity{Name='Tom', Age=84}

Entity dequeued: ExampleEntity{Name='Jackson', Age=76}

Entity dequeued: ExampleEntity{Name='Denis', Age=70}

Entity dequeued: ExampleEntity{Name='Violeta', Age=57}

Entity dequeued: ExampleEntity{Name='Eva', Age=35}

Entity dequeued: ExampleEntity{Name='Diana', Age=21}

Entity dequeued: ExampleEntity{Name='Viktoria', Age=21}

Higher age = higher priority

===============================================

Process finished with exit code 0

Код:

Launch:

public class Launch {  
 public static void main(String[] args) {  
 Demonstrator demonstrator = new Demonstrator();  
 String delimetr = "===============================================";  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstrateHashSet();  
 System.*out*.println(delimetr);  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstrateTreeSet();  
 System.*out*.println(delimetr);  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstrateTreeMap();  
 System.*out*.println(delimetr);  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstrateLinkedList();  
 System.*out*.println(delimetr);  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstrateArrayList();  
 System.*out*.println(delimetr);  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstrateQueue();  
 System.*out*.println(delimetr);  
  
 System.*out*.println(delimetr);  
 demonstrator.DemonstratePriorityQueue();  
 System.*out*.println(delimetr);  
 }  
}

public class Demonstrator {  
 private final String[] PossibleNames;  
 public Demonstrator(){  
 PossibleNames = new String[]{  
 "Roman", "Denis", "Tom", "Oleg", "Tom", "Jackson", "Michael",  
 "Viktoria", "Violeta", "Oleksandra", "Alina", "Eva", "Diana"  
 };  
 }  
 public Demonstrator(String[] possibleNames) {  
 PossibleNames = possibleNames;  
 }  
  
 private ExampleEntity GenerateExampleEntityValue(){  
 Random random = new Random();  
  
 int randomNameIndex = random.nextInt(PossibleNames.length);  
 String Name = PossibleNames[randomNameIndex];  
 int Age = random.nextInt(100);  
  
 return new ExampleEntity(Name, Age);  
 }  
 public void DemonstrateHashSet(){  
 System.*out*.println("HashSet Demonstration\n");  
 HashSet<ExampleEntity> entityHashSet = new HashSet<>();  
 ExampleEntity generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf(  
 "Proving, that HashCode generates same value for same objects: \n" +  
 "Entity: %s\nFirst gen: %d\n" +  
 "Second gen: %d\n\n",  
 generatedEntity, generatedEntity.hashCode(), generatedEntity.hashCode());  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: %s\tWith HashCode: %d\n", generatedEntity, generatedEntity.hashCode());  
 entityHashSet.add(generatedEntity);  
 }  
  
 System.*out*.printf("\nHashSet values: \n%s\n", entityHashSet);  
 }

public void DemonstrateTreeSet(){  
 System.*out*.println("TreeSet Demonstration\n");  
 TreeSet<ExampleEntity> entityTreeSet = new TreeSet<ExampleEntity>();  
 ExampleEntity generatedEntity;  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: %s\n", generatedEntity);  
 entityTreeSet.add(generatedEntity);  
 }  
 System.*out*.printf("\nGenerated TreeSet: \n%s\n", entityTreeSet);  
  
 ExampleEntity firstEntity = entityTreeSet.first();  
 ExampleEntity secondEntity = entityTreeSet.higher(firstEntity);  
 System.*out*.printf("\ncompareTo demonstration: \n" +  
 "(comparing by age)\n" +  
 "%s compareTo %s\nResult: %d\n",  
 firstEntity, secondEntity, firstEntity.compareTo(secondEntity));  
}  
public void DemonstrateTreeMap(){  
 System.*out*.println("TreeMap Demonstration\n");  
 TreeMap<Integer, ExampleEntity> entityTreeMap = new TreeMap<>();  
  
 ExampleEntity generatedEntity;  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: Key: %d \t Value: %s\n", i, generatedEntity);  
 entityTreeMap.put(i, generatedEntity);  
 }  
 System.*out*.printf("\nGenerated TreeMap: \n%s\n", entityTreeMap);  
  
 System.*out*.printf("\nValue with key='3': %s\n", entityTreeMap.get(3));  
}  
  
public void DemonstrateLinkedList(){  
 System.*out*.println("LinkedList Demonstration\n");  
 LinkedList<ExampleEntity> entityLinkedList = new LinkedList<>();  
  
 ExampleEntity generatedEntity;  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: %s\n", generatedEntity);  
 entityLinkedList.add(generatedEntity);  
 }  
  
 System.*out*.printf("\nGenerated LinkedList: \n%s\n", entityLinkedList);  
}

public void DemonstrateArrayList(){  
 System.*out*.println("ArrayList Demonstration\n");  
 ArrayList<ExampleEntity> entityArrayList = new ArrayList<>();  
 ExampleEntity generatedEntity;  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: %s\n", generatedEntity);  
 entityArrayList.add(generatedEntity);  
 }  
  
 System.*out*.printf("\nGenerated ArrayList: \n%s\n", entityArrayList);  
 }  
  
 public void DemonstrateQueue() {  
 System.*out*.println("Queue Demonstration\n");  
 Queue<ExampleEntity> entityQueue = new LinkedList<>();  
 ExampleEntity generatedEntity;  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: %s\n", generatedEntity);  
 entityQueue.add(generatedEntity);  
 }  
  
 System.*out*.println();  
  
 while (!entityQueue.isEmpty()) {  
 System.*out*.printf("Entity dequeued: %s\n", entityQueue.poll());  
 }  
  
 System.*out*.println();  
 }  
  
 public void DemonstratePriorityQueue() {  
 System.*out*.println("Queue Demonstration\n");  
 Queue<ExampleEntity> entityPriorityQueue = new PriorityQueue<>();  
 ExampleEntity generatedEntity;  
  
 for(int i = 0; i < 10; i++){  
 generatedEntity = GenerateExampleEntityValue();  
 System.*out*.printf("Entity created: %s\n", generatedEntity);  
 entityPriorityQueue.add(generatedEntity);  
 }  
  
 System.*out*.println();  
  
 while (!entityPriorityQueue.isEmpty()) {  
 System.*out*.printf("Entity dequeued: %s\n", entityPriorityQueue.poll());  
 }  
  
 System.*out*.println("\nHigher age = higher priority");  
 }  
}

package org.example.Lab6;  
  
import java.util.Objects;  
  
public class ExampleEntity implements Comparable<ExampleEntity>{  
 private String Name;  
 private int Age;  
  
 public ExampleEntity(String name, int age) {  
 Name = name;  
 Age = age;  
 }  
  
 public String getName() {  
 return Name;  
 }  
  
 public void setName(String name) {  
 Name = name;  
 }  
  
 public int getAge() {  
 return Age;  
 }  
  
 public void setAge(int age) {  
 Age = age;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 ExampleEntity that = (ExampleEntity) o;  
 return Age == that.Age && Name.equals(that.Name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(Name, Age);  
 }  
  
 @Override  
 public String toString() {  
 return "ExampleEntity{" +  
 "Name='" + Name + '\'' +  
 ", Age=" + Age +  
 '}';  
 }  
  
 @Override  
 public int compareTo(ExampleEntity o) {  
 if(this.Age < o.Age)  
 return 1;  
 else if (this.Age == o.Age)  
 return 0;  
 return -1;  
 }  
}